Package 'CIhdq'

December 11, 2019

Type Package

Title Regularized projection score estimation of treatment effects in high-dimensional quantile regression

Version 1.0.1

Date 2019-12-11

Author Xingdong Feng [aut],
Jian Huang [aut],
Xu Liu [aut,cre]

Maintainer Xu Liu <liu.xu@shufe.edu>

Depends R (>= 3.1.0)

Imports glmnet, grpreg, SparseM, quantreg

Description A regularized projection score method is proposed for estimating treatment effects in quantile regression in the presence of high-dimensional confounding covariates. This method is based on an estimated projection score function of the low-dimensional treatment parameters in the presence of high-dimensional confounding covariates. We propose one-step algorithm and a reffitted wild bootstrapping approach for variance estimation. This enables us to construct confidence intervals for the treatment effects in the high-dimensional circumstances.

License GPL-2

LazyData true

NeedsCompilation yes

Repository GitHub

URL https://github.com/xliusufe/CIhdq

Encoding UTF-8

R topics documented:

CIhdq-package	*Regularized projection score estimation of treatment effects in high-dimensional quantile regression*

Description

A regularized projection score method is proposed for estimating treatment effects in quantile regression in the presence of high-dimensional confounding covariates. This method is based on an estimated projection score function of the low-dimensional treatment parameters in the presence of high-dimensional confounding covariates. We propose one-step algorithm and a refitted wild bootstrapping approach for variance estimation. This enables us to construct confidence intervals for the treatment effects in the high-dimensional circumstances.

Details

Package:	CIhdq
Type:	Package
Version:	1.0.1
Date:	2019-12-12
License:	GPL-2

References

Feng, X., Huang, J. and Liu, X. (2019). Regularized projection score estimation of treatment effects in high-dimensional quantile regression. Manuscript.

---

inferen	*Provide CI of individual coefficient of high-dimensional quantile regression.*

Description

This function provides the confidence intevals of individual coefficient of high-dimensional quantile regression by a regularized projection score method for estimating treatment effects. One-step estimation procedure can speed up computation, and the Bootstrap method can narrow the length of CI.

Usage

```
inferen(y,x,z,tau,method="OneStep",pen="glasso",eps=1e-6,
        level=0.85,iter.num=100,RCV=F,K=1,weights=NULL,B=100)
```

Arguments

y	The response, a vector of size n
x	The treatment effects, a matrix with dimension $n \times p$
z	The confounders a matrix with dimension $n \times q$
tau	The given quantile, a scale in the unit inteval
method	The method including "OneStep", "Iterative". "OneStep" denotes one-step method (Feng et al. 2019); "Iterative" denotes that the iteration stops when algorithm conveges. Default is "OneStep".
pen	The penalty including "glasso" and "lasso". "glasso" denotes the grouped lasso that is used in the regression of treatment effect on confounders; "lasso" denotes the lasso. Default is "glasso".
eps	The perturbation when the proposed algorithm is used. Default is epsilon=1e-6.
level	The length of tuning parameter α which is selected automatically. Default is 50.
iter.num	The number of folds for the tuning selection by CV. Default is 5.
RCV	Use refitted cross validation method and wild bootstrap to estimate the asymptotic covariance matrix. Default is False.
K	The number of repeated RCV. Default is 1.
weights	The weights used for wild bootstrap; if not specified (=NULL). Default is NULL.
B	The size for bootstrap. Default is 1000.

Value

est	Estimator of β. It is a list.
cov	Covariance matrix of β. It is a $d \times d$-matrix.

Author(s)

Feng Xingdong, Huang, Jian and Xu Liu (liu.xu@sufe.edu.cn)

References

Feng, X., Huang, J. and Liu, X. (2019). Regularized projection score estimation of treatment effects in high-dimensional quantile regression. Manuscript.

Examples

```
n <- 50
d <- 3
s <- 3
p <- 20
beta <- rep(3,d)
eta <- c(rep(3,s),numeric(p-s))
x <- matrix(rnorm(n*d),n,d)
z <- matrix(rnorm(n*(p-1)),n,p-1)
y <- x%*%beta + cbind(1,z)%*%eta + rnorm(n)
fit <- inferen(y,x,z,tau=0.5)
ests <- fit$result
est.coef <- ests$coef
boot.var <- diag(fit$cov)
lbounds <- ests$coef - qnorm((1+alpha)/2)*sqrt(boot.var)
```

```
ubounds <- ests$coef + qnorm((1+alpha)/2)*sqrt(boot.var)
counts <- ifelse(lbounds<beta&beta<ubounds,1,0)
coverage <- cbind(coverage,counts)
```

Index